

**OBJECT ORIENTED APPARATUS AND METHOD FOR ALLOCATING
OBJECTS ON AN INVOCATION STACK IN A PARTIAL
COMPILATION ENVIRONMENT**

ABSTRACT OF THE DISCLOSURE

5 An object oriented mechanism and method allow allocating Java objects on a
method's invocation stack in a partial compilation environment under certain conditions.
Only the classes that are visible are taken into account when performing escape analysis
in accordance with the preferred embodiments. In a first aspect of the invention,
conservative assumptions are made to assure that objects are only allocated on an
10 invocation stack when this can be proven safe by examining only those classes in the
compilation unit. In a second aspect of the invention, the concept of visible classes is
extended to include other classes that may be found from a user-defined classpath that
matches the anticipated run-time classpath used to find classes during program execution.
When stack allocation decisions for a method depends on such classes that are outside the
15 compilation unit, two versions of run time code for that method are created. One version
allocates all objects from the heap, whereas the other allocates some or all objects on the
invocation stack. For each class outside the compilation unit that was examined when
making stack allocation decisions for a method, information about the requisite class is
stored with the method. At run time, the class loader verifies that the run time version of
20 each requisite class matches all stored information about that class. If all information
matches for all requisite classes, the version of the method using stack allocation is safe
to use at run time. Otherwise, the method using heap allocation must be used.